BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

As rescanning documents *will not* correct images, please do not report the images to the Image Problem Mailbox.

(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) Publication number: **0 371 229 A2**

(12) EUROPEAN PATENT APPLICATION

(21) Application number: **89117755.2**

(22) Date of filing: **26.09.89**

(51) Int. Cl.[5]: **G06F 15/16, G06F 15/40, G06F 9/46**

The title of the invention has been amended (Guidelines for Examination in the EPO, A-III, 7.3).

(30) Priority: **31.10.88 US 265421**

(43) Date of publication of application: **06.06.90 Bulletin 90/23**

(84) Designated Contracting States: **DE FR GB IT**

(71) Applicant: **Hewlett-Packard Company**
3000 Hanover Street
Palo Alto California 94304(US)

(72) Inventor: **Blakely, Frank W.**
1404 Foxglove Court
Roseville, CA 95661(US)
Inventor: **Hall, Guy T.**
8325 Walden Woods Way
Loomis, CA(US)

Inventor: **Winkleblack, Sherry**
8701 Country Creek Drive
Orangevale, CA 95662(US)
Inventor: **Scaccia, Jim**
1446 Spring Valley Drive
Roseville, CA 95661(US)
Inventor: **Iwamoto, Shinichi**
777-4 Minaminakamaru
Ohmiya-shi, Saitama-Ken(JP)
Inventor: **Nojiri, Minoru**
4-3-17 Sakurayama, Zushi-shi
Kanagawa-Ken(JP)
Inventor: **Umezawa, Yukihoko**
1568 9th Avenue
Sacramento, Ca 95818(US)

(74) Representative: **Liesegang, Roland, Dr.-ing. et al**
FORRESTER & BOEHMERT
Widenmayerstrasse 4
D-8000 München 22(DE)

(54) Computer system having host computer.

(57) A computer system is presented. The computer system has a host computer (20), one or more personal computers (10) and a data transport system (30,31) which transports data between the host computer (20) and the personal computers (10).

On each personal computer (10) resides a number of applications (11). Each application (11) may utilize the resources on the host computer (20) by use of a requester program (13,15) existing on the personal computer (10) and a host server (24) residing on the host computer (20). When the application (11) desires a command to be processed by the host server (24), the application (11) transfers the command to the requester program (13,15). The requester program (13,15) receives the command and any parameters or other data associated therewith. The requester program (13,15) translates or reformats the command and associated information in preparation for sending the command et al. to the host computer (20).

The requester program (13,15) hands the translated command to data transport software (17,18). The data transport software (17,18) serves to interface the requester program (13, 15) with the data transport system (30,31) and serves to interface the host server (24) with the transport system (30,31). The translated command is thereby transferred from the requester program (13,15) to the host server (24).

The host server (24) oversees the execution of the command. The host server (24) may call data base intrinsics (22), operating system intrinsics (21) or remote procedure intrinsics (23) to execute the command.

When the host server (24) is started in response to a request originated by an application (11), host server (24) allocates a certain amount of memory on the host computer (20) for use by the application

EP 0 371 229 A2

(11) as a scratch pad. Through the use of remote procedures (23) the application (11) may store information to the scratch pad, modify data in the scratch pad and retrieve data from the scratch pad.

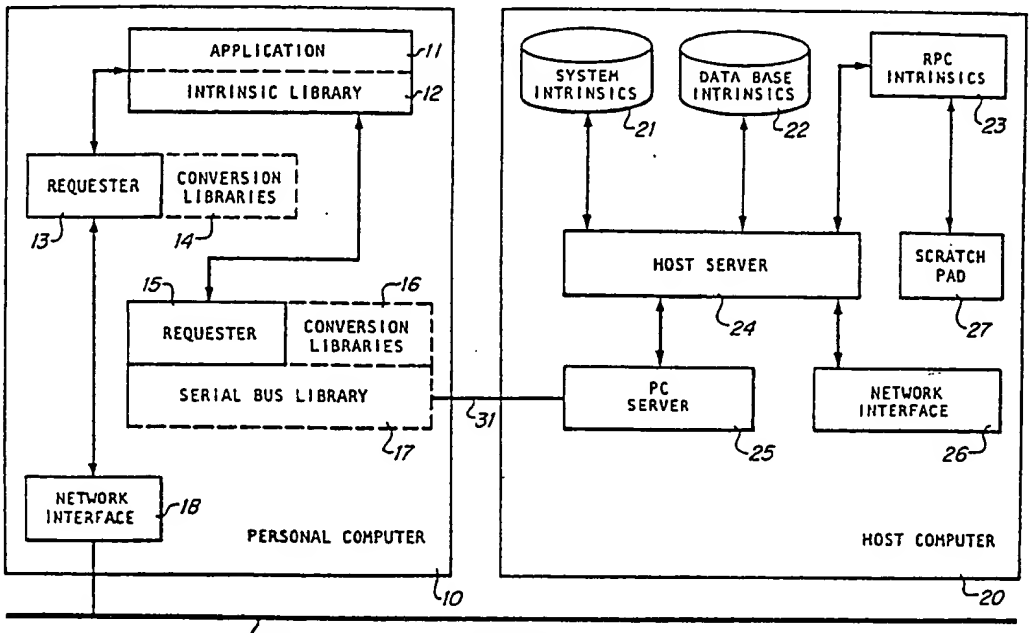

Figure 1

Computer System

Background

The use of personal computers for processing provides several advantages to the user. For instance, since there is typically only one user, computer response time is predictable. Further, a personal computer may be portable, and is isolated from other users so that there is no worry about another user causing a system "crash".

Similarly, performing operations on a larger, or host computer provides certain advantages. For instance, on a host computer a central data base may be kept. In such a data base a large amount of data may be centrally managed assuring that all users of the data base have access to valid, "up-to-date" information. It is desirable to design systems which utilize both personal computers and a host computer and which profit from the advantages of each type of computer.

Summary of the Invention

In accordance with the preferred embodiment of the present invention a computer system is presented. The computer system has a host computer, one or more personal computers and a data transport system which transports data between the host computer and the personal computers.

On each personal computer reside a number of applications. Each application may utilize the resources on the host computer by use of a requester program existing on the personal computer and a host server residing on the host computer. When the application desires a command to be processed by the host server, the application transfers the command to the requester program. Transfer of the command may be done in the form of a normal procedure call. The requester receives the command and any parameters or other data associated therewith. The requester program translates or reformats the command and associated information in preparation for sending the command et al. to the host computer.

The requester hands the translated command to data transport software. The data transport software serves to interface the requester with the data transport system and serves to interface the host server with the transport system. The translated command is thereby transferred from the requester to the host server.

The host server oversees the execution of the commands. In the preferred embodiment, there are five categories of commands. The first category includes commands which pertain to the setting up and maintaining of a session between the host server and the requester. Once the host server is set up, commands in this category are typically executed by the host server. The second category of commands includes commands which request interaction with a data base. Host server typically calls data base intrinsics to execute these commands. The third category of commands includes commands which are calls to the operating system. Host server typically calls operating system intrinsics to execute these commands. The fourth category of commands includes commands which are calls to remote procedures. These remote procedures are procedures written for a specific application, but which reside on the host computer rather than the personal computer. The host server typically call remote procedure intrinsics to execute these commands. The fifth category of commands includes commands which are used for housekeeping. For example, the requester may request host server for certain status information. Commands in this category are typically executed by the host server.

When the host server is started in response to a requests originated by an application, host server allocates a certain amount of memory on the host computer for use by the application as a scratch pad. Through the use of remote procedures the application may store information to the scratch pad, modify data in the scratch pad and retrieve data from the scratch pad.

Brief Description of the Drawings

Figure 1 shows program modules within a personal computer and within a host computer which allow for cooperative processing between the host computer and personal computer in accordance with a preferred embodiment of the present invention.

Figure 2 shows an alternate embodiment of some of the program modules shown in Figure 1.

Figure 3 shows a parameter block and details of an intrinsic library shown in Figure 1.

Figure 4 shows additional detail of the program modules shown in Figure 1.

Figure 5 shows a block diagram of messages sent between the host computer and the personal computer shown in Figure 1 in accordance with the preferred embodiment of the

present invention.

Description of the Preferred Embodiment

Figure 1 shows program modules within a personal computer 10 and a host computer 20. These program modules allow for cooperative processing between host computer 20 and personal computer 10. An application 11 calls an intrinsic library 12 in the same way application 11 makes other procedure or function calls. For example, this means that a return address, parameter values and/or addresses are placed on a program stack segment 70 (shown in Figure 3), and that control is transferred to intrinsic library 12. In the preferred embodiment host computer 20 is an HP 3000 computer available from Hewlett Packard Company, a California Corporation having a place of business at 3000 Hanover Street, Palo Alto, California 94304. In the preferred embodiment personal computer 10 may be, for example a VECTRA personal computer, a Touchscreen personal computer, or a Portable Plus personal computer all available from Hewlett Packard Company.

As shown in Figure 3, intrinsic library 12 is composed of 5 modules. Each module contains intrinsics which are responsible for processing a particular class of intrinsic calls. Each module places output into parameter block 60. Parameter block 60 receives from each module two words 61 of status information, and a number of words 62 of parameter values and addresses. For each module the number and arrangement of words 62 varies.

The first module within intrinsic library 12 is session management intrinsics module 51. Session management intrinsics module 51 handles commands from application 11 which are directed to control the link between personal computer 10 and host computer 20 and which perform other tasks which manage the session between personal computer 10 and host computer 20. Session management intrinsics module 51 extracts information from program stack segment 70 and after changing the format, places the information in parameter block 60.

Data base intrinsics module 52 handles commands from application 11 which are to be executed by data base intrinsics 22. Data base intrinsics 22 resides on host computer 20 as shown in Figure 1. With the use of a table, data base intrinsics module 52 generates output which is placed in parameter block 60. Using the table, data base intrinsics module takes parameters and other information from program stack 70, translates the parameters and other information, and places them in parameter block 60.

Translation of the parameters may include, for example, placing the parameters in parameter block 60 without modifying the parameters, modifying the parameters before placing them in parameter block 60, or placing substitute parameters or other information into parameter block 60.

Data base intrinsics module 52 also uses the table to determine how parameters are to be passed to host computer 20. For example, the implemented data base may be TurboIMAGE, available from Hewlett Packard Company. System intrinsics module 53 handles commands from application 11 which are to be executed by system intrinsics 21. System intrinsics 21 reside on host computer 20. With the use of a table, system intrinsics module 53 generates output for placement in parameter block 60. Using the table, system intrinsics module 53 takes parameters and other information from program stack 70, reformats the parameters and other information, and places them in parameter block 60. System intrinsics module also uses the table to determine how a parameters are to be passed to host computer 20. In the preferred embodiment of the present invention MPE is used as the system program for host computer 20. MPE system program for HP 3000 computer is available from Hewlett Packard Company.

Remote procedure call (RPC) intrinsics module 54 handles commands from application 11 which are to be executed by RPC intrinsics 23. RPC intrinsics 23 reside on host 20. With the use of a table, RPC intrinsics module 54 generates output for placement in parameter block 60. Using the table, RPC intrinsics module 54 takes parameters and other information from program stack 70, reformats the parameters and other information, and places them in parameter block 60.

Conversion intrinsics module 55 handles commands from application 11 which are to be executed by conversion modules 14 or conversion modules 16. Conversion intrinsics module 55 extracts information from program stack 70, reformats the information and places the information in parameter block 60.

As shown in Figure 1, requester 15 receives intrinsic calls from intrinsic library 12 when data is to be transferred to host 20 through a point to point data communication link 31. Requester 13 receives intrinsic calls from intrinsic library 12 when data is to be transferred to host 20 through a local area network (LAN) 30. When requester 13 or requester 15 receives a software interrupt from intrinsic library 12, requester 13 or requester 15 accesses a predetermined address to discover the location within parameter block 60 which contains addresses and parameters related to the particular procedure call.

Requester 13 and Requester 15 have the same

structure. In Figure 4, requester 13 is shown to be composed of five modules: a session management intrinsics module 81 which processes session management intrinsics, a data base intrinsics module 82 which processes data base intrinsics, a system intrinsics module 83 which processes system intrinsics, a remote procedure call intrinsics module 84 which processes remote procedure call intrinsics and a conversion intrinsics module 85 which processes remote procedure call intrinsics. Each logic module extracts information from parameter block 60.

Session management intrinsics module 81, data base intrinsics module 82, system intrinsics module 83 and remote procedure call intrinsics module 84 utilize a "skeleton' processor'. The skeleton processor is essentially code which uses a table that defines the steps which must be taken to process a given intrinsic. Based on information in the table, the identity of the intrinsic and the parameters of the intrinsic the skeleton processor processes the intrinsic. Each intrinsic is processed using all or a subset of process steps. The process steps are common across all intrinsics, although a particular intrinsic may not use all of the steps.

Output from session management intrinsics module 81, data base intrinsics module 82, system intrinsics module 83 and remote procedure call intrinsics module 84 are placed in data communication buffer 90. Data placed in data communication buffer 90 is in the form of messages. The form of these messages is that of a message 100, shown in Figure 5. Parameter block 60 is used to generate the messages, however the data within the messages may differ significantly from the data within parameter block 60. For example, pointers to values in parameter block 60 are replaced by the values themselves in the generated messages. Conversion intrinsics module 85 passes conversion intrinsics to a appropriate conversion library within conversion libraries 14. A table determines which conversion library will be used and how the information will be reformatted.

In Figure 1, conversion libraries 14 and conversion libraries 16 consist of routines which convert specific data formats which are available on host computer 20 into an equivalent format for personal computer 10. These routines also convert specific data formats which are available on personal computer 10 into an equivalent format for host computer 20. This is necessary because different programming languages used may use different formatting for stored data. Also, the same programming languages may use different formatting for storing data when the programming language is run on a different type of computer.

If data communication link 31 is used for transmission of data between personal computer 10 and host computer 20, requester 15 calls serial bus library 17 to transfer information over data communication link 31 to a PC server 25 within host computer 20. Standard point to point data communication technology may be used. For example, serial bus library 17 may be implemented using Extended Data Communication Library (EDCL) available as a product from Hewlett Packard Company with a product name of Basic Serial. Also, PC server 25 may be implemented using PCSERVER, which is part of the fundamental Operating System distributed for use on Hewlett Packard Company Model 3000 computers.

If LAN 30 is used for transmission of data between personal computer 10 and host computer 20, requester 13 calls network interface 18 for the transfer of information over LAN 30 to a network interface 26 within host computer 20. Standard LAN technology may be used to implement Network interface 18 and network interface 26. For instance, network interface 18 may be implemented using OfficeShare available from Hewlett Packard Company. Similarly, network interface 26 may be implemented using AdvanceNet, also available from Hewlett Packard Company. A host server 24 receives messages from PC server 25 and network interface 26 processes the messages, and returns messages to requester 13 and requester 15. Host server 24 may consist of two separate programs: one program which handles messages which come over serial data communication link 31 and a separate program which handles messages which come over LAN 30.

In order to execute the messages, host server 24 calls system intrinsics 21, data base intrinsics 22 and RPC intrinsics 23. Depending upon the type of software such as system software and data base software used, different intrinsics may be implemented. For instance, in the preferred embodiment, data base intrinsics 22 include intrinsics for the IMAGE data base which execute the following commands: DBOPEN, DBINFO, DBCLOSE, DBFIND, DBGET, DBUPDATE, DBPUT, DBDELETE, DBLOCK, DBUNLOCK, DBCONTROL, DBBEGIN, DBEND, DBMEMO, DBERROR and DBEXPLAIN. Similarly, system intrinsics 21 execute the following MPE FILE commands: FCHECK, FCLOSE, FCONTROL, FFILEINFO, FOPEN, FREAD, FWRITE, FERRMSG and COMMAND.

RPC intrinsics 21 are host routines written, compiled, and placed in host libraries which can be called by host server 24 upon requests sent from personal computer 10. A host routine, once called, uses data provided in a request message originated from personal computer 10. The host routine in a return message returns data and completion status to personal computer 10.

Upon being started, host server 24 allocates a

portion of memory as a scratch pad 27. Scratch pad 27 may be used by routines on personal computer 10 as temporary local storage on host 20. For example, application 11 through RPC intrinsics 23 may store, modify and retrieve data placed in scratch pad 27. Upon termination of host server 24, the memory allocated to scratch pad 27 is freed by host 20 for reallocation.

Additional commands are executed by intrinsics implemented in the preferred embodiment of the present invention. For example, a "trace on" and a "trace off" command are implemented. The "trace on" command directs host server 24 to log data communication and/or intrinsic tracing data to a file. Also, a "start session" and a "stop session" are implemented. These commands direct host server 24 to start or stop a server session on host 20. Further, a "status checking" command are implemented. A "status checking" command, generated by requester 13 or by requester 15, causes host server 24 to verify compatible interface versions are being used. Finally, a READ STDLIST command directs host server 24 to read a standard list device output generated by DBEXPLAIN (see above) for transmission back to personal computer 10.

Figure 5 shows the format of messages sent between personal computer 10 to host computer 20. On personal computer 10, requester 13 and requester 15 create and interpret the messages. On host computer 20, host server 24 creates and interprets the messages.

As shown in Figure 5, a message 100 includes a packet header 101 and a packet body 102. Packet header 101 includes a packet number 110, a product ID 111, packet attribute byte 112 and a packet status byte 113. Packet attribute byte indicates the origin of the message, e.g. whether it is from host 20 or personal computer 10.

Packet body 102 includes an operation category byte 114 and a function type 115. For example, operation category byte 114 may indicate in the category of an intrinsic included in the message. For instance, the intrinsic may be categorized as (1) system control intrinsic, (2) a data base intrinsic, (3) an operating system intrinsic, (4) a remote procedure call intrinsic, or (5) a special intrinsic used for housekeeping, status, etc. Function type 115 may contain or identify a command which invokes a particular intrinsic.

When application 11 desires to start a session on host computer 11, application makes two function calls: a function call requesting a connection to host computer 20, and a function requesting the start of a session. Requester 13 and requester 15 respond differently to these two commands.

Upon receiving a connect function call, requester 15 sends out a message containing a "log on" string to PC server 25. PC server 25 passes the "log on" string to the operating system of host computer 20. Upon receiving from application 11 a request for a start session, requester 15 directs PC server 25 to start host server 24.

PC server 25 starts host server 24 by issuing a create process intrinsic. Upon starting, host server 24 determines how it was initiated. Upon determination that PC server 25 initiated host server 24, host server 24 opens two circular message files. These are used for communication between PC server 25 and host server 24. Host server 24 then waits for a check status message from requester 15. The check status message allows host server 24 to determine whether host server 24 and requester 15 are compatible. Once it has been determined that compatible interface versions are being used, host server 24 proceeds to process messages from personal computer 10 until host server 24 receives a session stop message. At that point the two circular message files are closed and host server 24 terminates.

Upon receiving a connect function call, requester 13 sends out a message requesting network interface 26 to start a host server session. Network interface 26 starts host server 24 by use of a system process. For example, in the preferred embodiment, when AdvanceNet is used, system process DSDAD through a network monitor process monitors LAN 30. Upon receiving the request to start a server session, DSDAD starts host server 24.

Upon starting, host server 24 determines how it was initiated. Upon determination that a system process initiated host server 24, host server 24 creates a port and adds it to a list of open ports within host 20. Host server 24 then waits for a check status message and a start session message. When a start session message is received, host server 24 gets a virtual terminal connection, starts a session using a "log on" string from requester 13. Host server 24 then adopts itself to the started session, that is, host server 24 causes its father process to be changed from DSDAD to the started session. Host server 24 then processes messages received from personal computer 10 until a stop session message is received. Then host server 24 terminates the sessions and closes the virtual terminal connection.

Messages from personal computer 10 to host computer 20 carry commands which are executed by system intrinsics 21, data base intrinsics 22 and RPC intrinsics 23. Generally, after execution, host computer 20 returns to personal computer 10 a return message which includes, for example, data and/or status.

For messages carrying commands for IMAGE data base to be executed by data base intrinsics

22, data 116 contains parameters to the file and command intrinsics, and where a parameter is variable in length, a size parameter. Messages returned to personal computer 10 consists of a condition code, status and data appropriate for the intrinsic. However, return messages for IMAGE commands DBERROR and DBEXPLAIN do not include status. Further DBEXPLAIN does not return a message but writes information into a storage location IMAGE called a stdlist file. In order to retrieve this information personal computer 10 issues a READ STDLIST command to get a message.

For messages carrying commands for an MPE system program to be executed by system intrinsics 21, data 116 contains parameters to the file intrinsics--which are the intrinsics which execute FREAD and FOPEN commands--and to the COMMAND intrinsic and, where a parameter is variable length, a size parameter. Messages returned to personal computer 10 include a condition code and the data normally returned to a process which executes the particular MPE system command.

The FREAD and FOPEN intrinsics return messages which have an extra parameter. The parameter returns function values from the intrinsic.

For messages which carry remote procedure calls to be executed by RPC intrinsics 23, data 116 contains RPC data, a field designating the number of bytes of RPC data being sent and a field designating the number of bytes of return data personal computer 10 expects to receive from host computer 20 as a result of the message. Messages returned to personal computer 10 include a condition code, status, return data and a field designating the number of bytes of return data. The RPC data includes the name of the procedure to be called and the data to be passed to the procedure. Upon receiving a message for a remote procedure call, host server 24 retrieves the procedure from RPC intrinsics 23, executes the intrinsic and loads the return data into the return message.

Host server 24 recognizes a "trace on" message and a "trace off" message. The request message for a "trace on" contains fields which indicate the type of tracing to be done. This field indicates whether intrinsic tracing, data communication tracing, or both is to occur.

When host server 24 receives a message indicating intrinsic tracing is to be done, host server 24 opens an output file. Into this output file is logged the data from request message. Also placed in this file is return data from return messages sent in response to messages which carry remote procedure calls.

When host server 24 receives a message indicating data communication tracing is to be done, host server 24 also opens an output file. Into this file is logged data communications packets which pass between PC server 25 and host server 24, and/or data communication packets which pass between network interface 26 and host server 24.

When host server 24 receives a "trace off" message no further logging is done and any open output files are closed.

Host server 24 also recognizes a "start session" message and a "stop session" message which are sent over LAN 30. For a "start session" message, data 116 contains a character string used for "log on" to host computer 24. A return message for the "start session" message includes information pertaining to any detected error. For a "stop session" message no data is sent. A return message for the "stop session" message includes information pertaining to any detected error.

For a "start session" message sent over LAN 30, host server 24 gets a virtual (pseudo) terminal connection, starts a session using the included character string and adopts itself to that session. For a "stop session" message sent over LAN 30, host server 24 terminates the session and closes the virtual terminal connection.

Requester 13 or requester 15 to host server 24 can send a request message which includes a "status checking" command. Such a request message includes within data 116 the packet buffer size, version number of requester 13 or 15, and a transport flag, which gives further information on capabilities supported by the requester initiating the message. A return message sent from host server 24, includes the error information and the version number of host server 24.

Upon receiving a request message which includes a "status checking" command host server 24 checks the enclosed version number to determine whether the version number sent in the message describes a version host server 24 is able to interact with. If so host server 24 returns its own version number to personal computer 10. Otherwise, the return message includes error information.

A request message for a READ STDLIST command does not contain data field 116. A return message sent from host 24 includes error information, data, and a field which indicates the amount of data within the return message.

Upon receipt of a request message for a READ STDLIST command, host server 24 closes the stdlist file, and loads the contents into the return message.

In Figure 2, a personal computer 40 is shown to have an alternate configuration to personal computer 10. An application 41 calls an intrinsic library 42 in the same way application 41 makes other procedure or function calls. A single requester 43 receives intrinsic calls from intrinsic library 42 when data is to be transferred to host 20 through

data communication link 31 or LAN 30. Requester 43 calls serial bus library 45 to transfer information over data communication link 31 to PC server 25 within host computer 20. Requester 43 calls network interface 45 for the transfer of information over LAN 30 to network interface 26 within host computer 20. Conversion intrinsics from intrinsic library 42 are passed to conversion libraries 44.

Appendix A contains source code listings of network interface 26, requester 13 and host server 24 in accordance with the preferred embodiment of the present invention.

Claims

1. Computer system having at least one first computer (10), host computer (20) and a data transport system (30, 31) which is coupled to the first computer (10) and the host computer (20), comprising a plurality of program segments which facilitate the execution of a plurality of commands by the host computer, the plurality of commands being originated by an application (11) which resides on the first computer (10), **characterized** in that the program segments comprise:
a requester program (13, 15) residing on the first computer (10), which receives commands from the application (11) and in preparation for transport of the commands, translates the commands to produce translated commands;
data transport software (17, 18) residing within the first computer (10) and within the host computer (20), the data transport software receiving the translated commands from the requester program and causing the data transport system to transport the translated commands to the host computer;
a host server program (24), residing on the host computer (20), the host server program receiving the translated commands and overseeing the execution of the commands; and,
intrinsic program segments (21, 22, 23) residing on the host computer (20), the intrinsic program being called by the server program (24) in order to execute the commands.

2. Computer system as in claim 1, **characterized** in that the intrinsic program segments include:
data base intrinsic program segments (22) which are called by the host server program to execute commands which operate on a data base;
system intrinsic program segments (21) which are called by the host server program to execute commands which are commands normally executed by an operating system of the host computer; and,
remote procedure intrinsic program segments (23) which are called by the host server program to execute commands specific to the application program.

3. Computer system as in claim 2, **characterized** in that a segment of memory is reserved by the host server (24) as a scratch pad (27) for the application program to use as temporary storage on the host server, and wherein the remote procedure intrinsic program segments (23) include means for, in response to commands from the application, storing data in the scratch pad (27), modifiying data in the scratch pad, and retrieving data from the scratch pad.

4. Computer system as in one of claims 1 to 3, **characterized** in that the requester (13, 15) and the host server program (24) interact by means of a protocol, the protocol including a first message originated by the requester and a return message originated by the host server program, the first message including data which identifies the version of the requester and when the host server program is compatible with the requester program, the return message including data which identifies the version of the host server program.

5. Method for facilitating the execution of a plurality of commands by a host computer (20) in a computer system having a first compouter (10), the host computer (20) and a data transport system (30, 31) which is coupled to the first computer and the host computer, the plurality of commands being originated by an application which resides on the first computer (10), **characterized** by:

(a) transferring the commands from the application to a requester program segment (13, 15);

(b) preparing the commands, by the requester program segment, to be transported to the host computer (20) by translating the commands to produce translated commands;

(c) transporting the translated commands from the first computer (10) to the host computer (20) over the data transport system (30, 31);

(d) receiving of the translated commands by a host server program (24) residing on the host computer (20); and,

(e) calling for intrinsic program segments (21, 22, 23) by the host server program, to execute the commands.

6. The method as in claim 5, **characterized** in that step (e) additionally comprises the substeps of:

(e1) calling data base intrinsic program segments (22) for the execution of commands which operate on a data base;

(e2) calling system intrinsic program segment (21) for the execution of commands which are commands normally executed by an operating system of the host computer (20); and,

(e3) calling remote procedure intrinsic program segments (23) for the execution of commands specific to the application program (11).

7. The method as in claim 6, **characterized** by

the additional steps of:

(f) reserving, by the host server program (24), a portion of memory in the host as a scratch pad (27) for the application program to use as temporary storage on the host server; and

(g) performing the following substeps in response to commands from the application

(g1) storing data in the scratch pad,

(g2) modifying data in the scratch pad, and

(g3) retrieving data from the scratch pad.

8. A method as in claim 5 or 6, **characterized** by the additional steps of:

(f) sending a message from the requester (15) to the host server (24), the message including data which identifies the version of the requester, and

(g) when the host server program is compatible with the requester program, performing the following substep:

(g1) sending a return message from the host server program to the requester, the return message including data which identifies the version of the host server program.

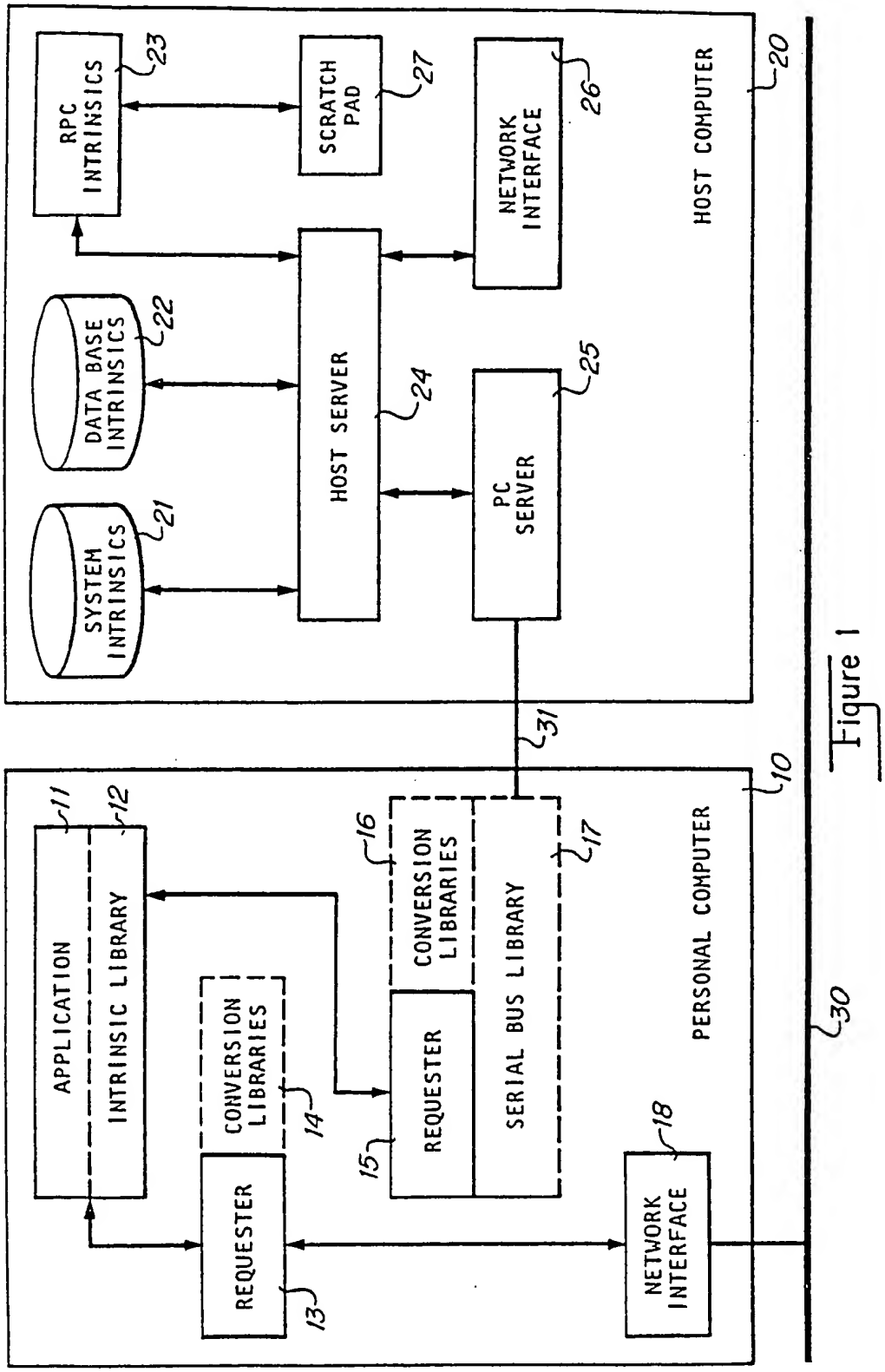

Figure 1

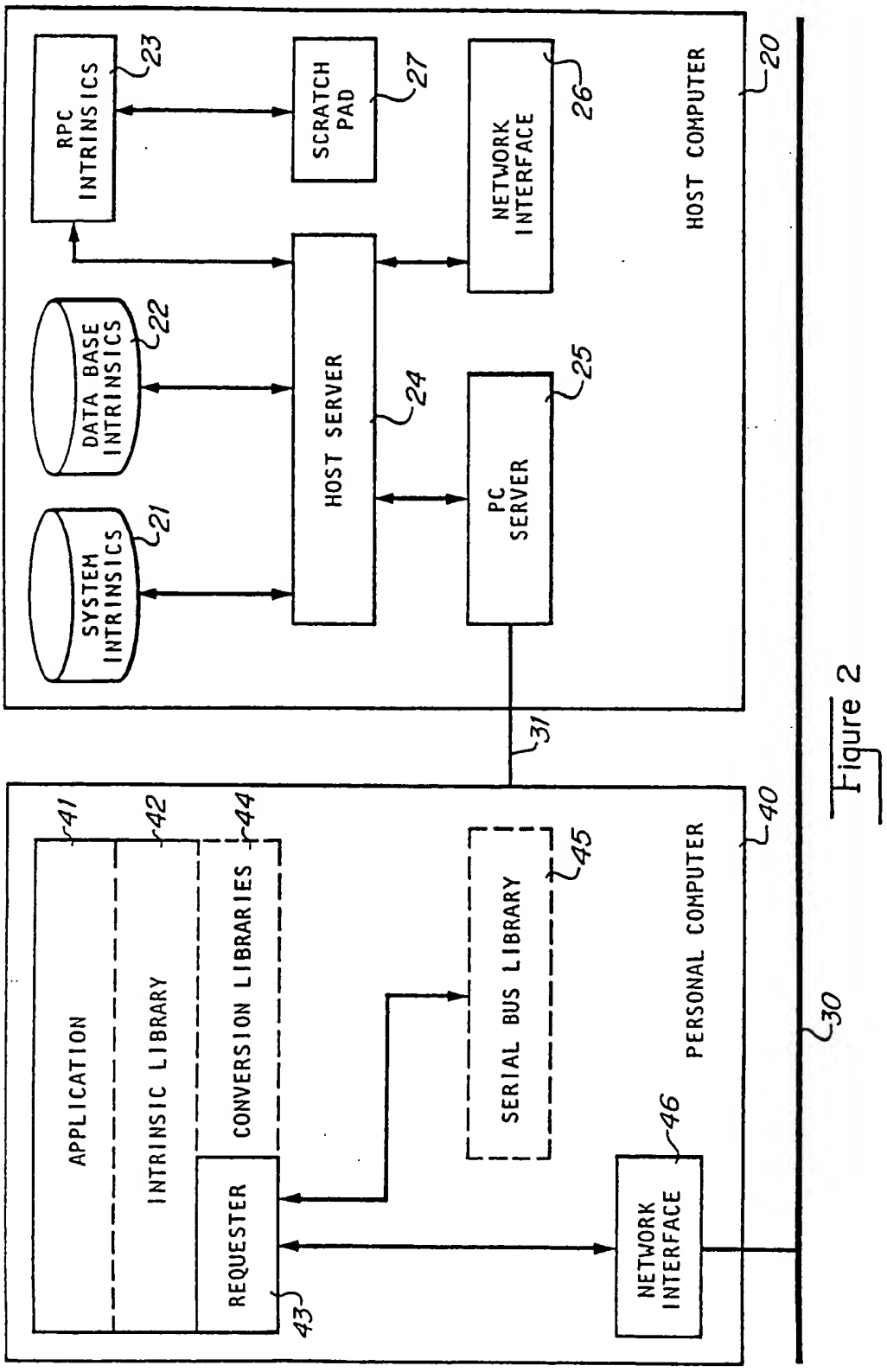

Figure 2

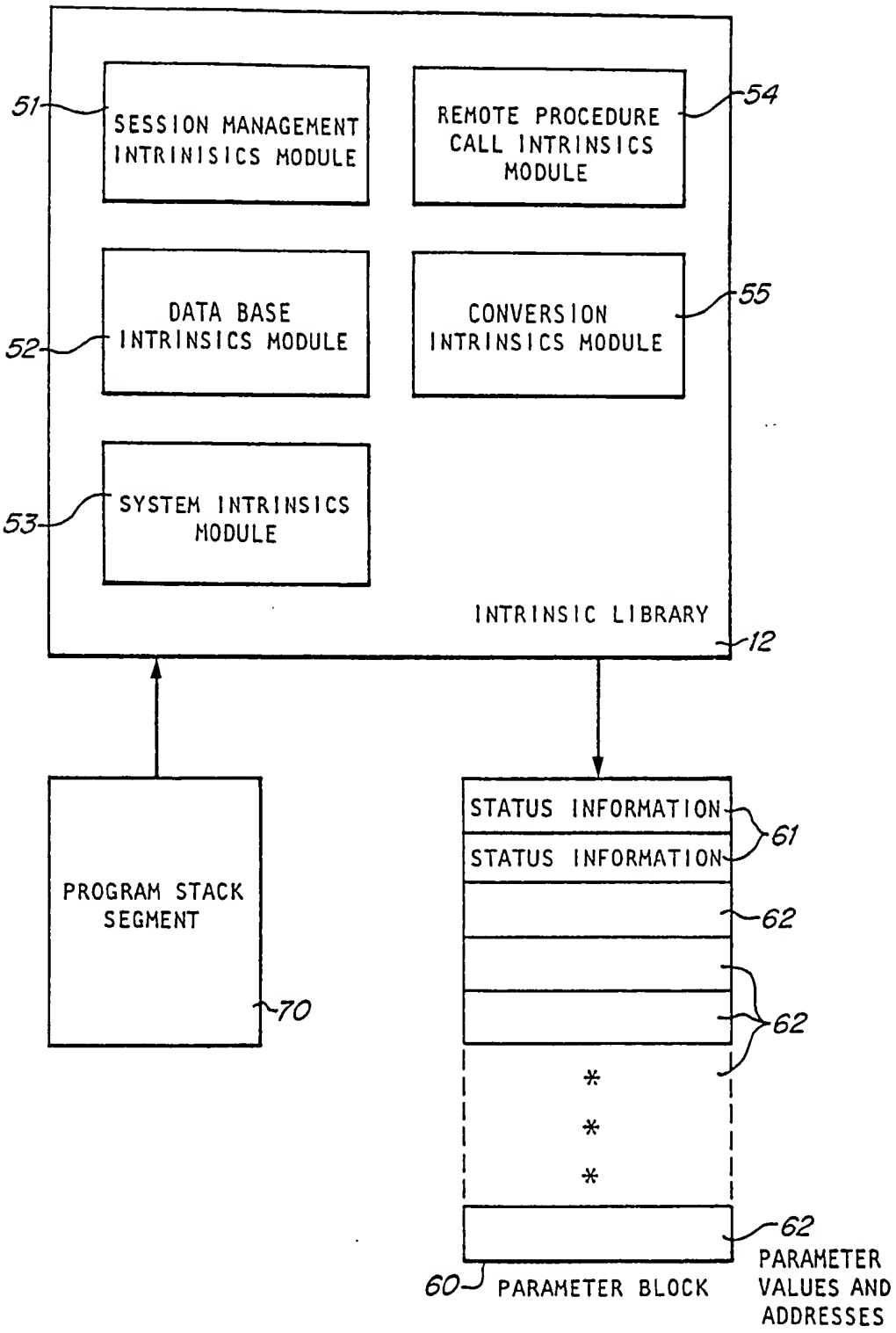
SESSION MANAGEMENT INTRINISICS MODULE
51
REMOTE PROCEDURE CALL INTRINSICS MODULE
54
DATA BASE INTRINSICS MODULE
52
CONVERSION INTRINSICS MODULE
55
SYSTEM INTRINSICS MODULE
53
INTRINSIC LIBRARY
12
PROGRAM STACK SEGMENT
70
STATUS INFORMATION
STATUS INFORMATION
61
62
62
62
60
PARAMETER BLOCK
PARAMETER VALUES AND ADDRESSES

Figure 3

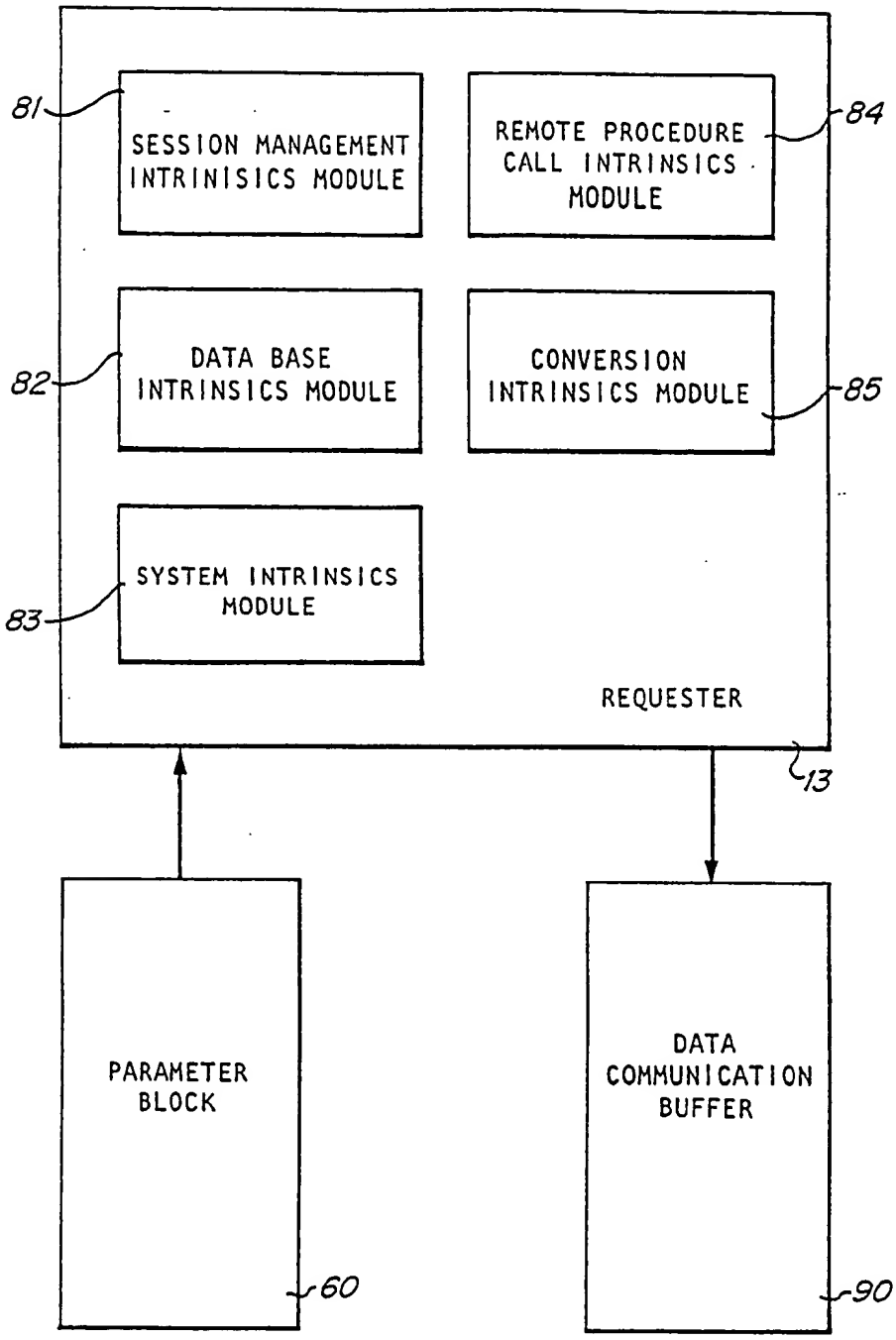

Figure 4

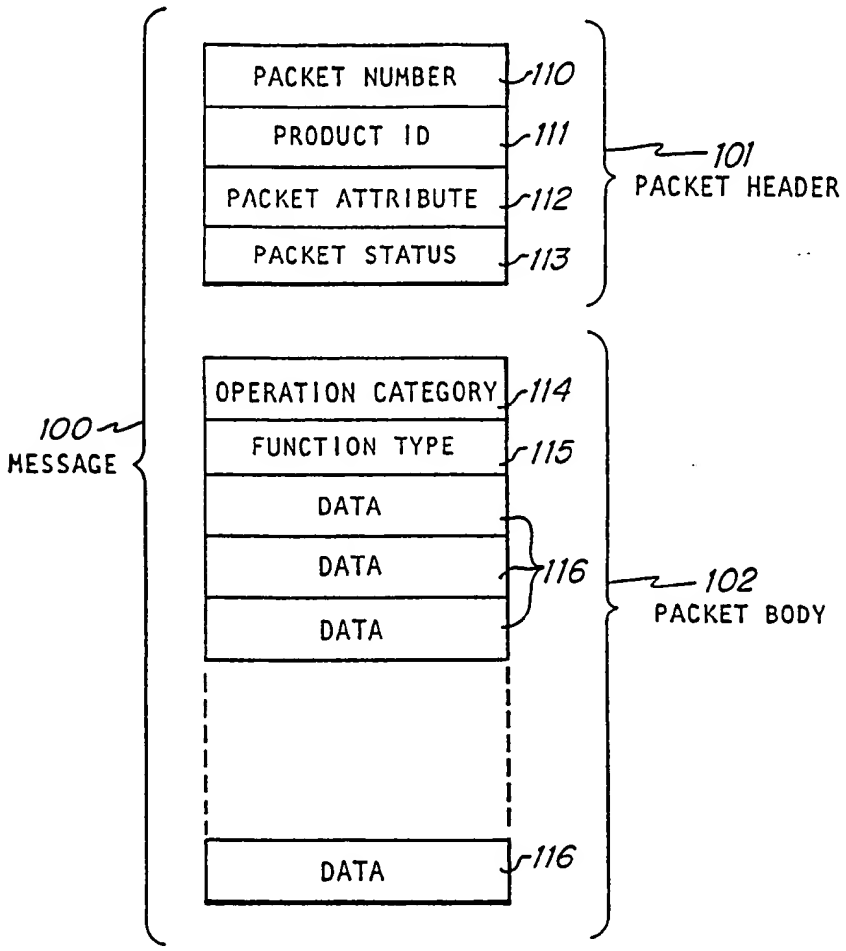
PACKET NUMBER
110
PRODUCT ID
111
PACKET ATTRIBUTE
112
PACKET STATUS
113
101
PACKET HEADER
OPERATION CATEGORY
114
FUNCTION TYPE
115
DATA
DATA
DATA
116
102
PACKET BODY
DATA
116
100
MESSAGE

Figure 5

(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) Publication number: **0 371 229 A3**

(12) **EUROPEAN PATENT APPLICATION**

(21) Application number: **89117755.2**

(22) Date of filing: **26.09.89**

(51) Int. Cl.5: **G06F 15/16**, G06F 15/40, G06F 9/46

(30) Priority: **31.10.88 US 265421**

(43) Date of publication of application: **06.06.90 Bulletin 90/23**

(84) Designated Contracting States: **DE FR GB IT**

(88) Date of deferred publication of the search report: **29.04.92 Bulletin 92/18**

(71) Applicant: **Hewlett-Packard Company**
Mail Stop 20 B-O, 3000 Hanover Street
Palo Alto, California 94304(US)

(72) Inventor: **Blakely, Frank W.**
1404 Foxglove Court
Roseville, CA 95661(US)
Inventor: **Hall, Guy T.**
8325 Walden Woods Way
Loomis, CA(US)
Inventor: **Winkleblack, Sherry**
8701 Country Creek Drive
Orangevale, CA 95662(US)
Inventor: **Scaccia, Jim**
1446 Spring Valley Drive
Roseville, CA 95661(US)
Inventor: **Iwamoto, Shinichi**
777-4 Minaminakamaru
Ohmiya-shi, Saitama-Ken(JP)
Inventor: **Nojiri, Minoru**
4-3-17 Sakurayama, Zushi-shi
Kanagawa-Ken(JP)
Inventor: **Umezawa, Yukihoko**
1568 9th Avenue
Sacramento, Ca 95818(US)

(74) Representative: **Liesegang, Roland, Dr.-Ing. et al**
FORRESTER & BOEHMERT
Franz-Joseph-Strasse 38
W-8000 München 40(DE)

(54) **Computer system having host computer.**

(57) A computer system is presented. The computer system has a host computer (20), one or more personal computers (10) and a data transport system (30,31) which transports data between the host computer (20) and the personal computers (10).

On each personal computer (10) resides a number of applications (11). Each application (11) may utilize the resources on the host computer (20) by use of a requester program (13,15) existing on the personal computer (10) and a host server (24) residing on the host computer (20). When the application (11) desires a command to be processed by the host server (24), the application (11) transfers the command to the requester program (13,15). The requester program (13,15) receives the command and any parameters or other data associated therewith. The requester program (13,15) translates or reformats the command and associated information in preparation for sending the command et al. to the host computer (20).

The requester program (13,15) hands the translated command to data transport software (17,18). The data transport software (17,18) serves to interface the requester program (13, 15) with the data transport system (30,31) and serves to interface the host server (24) with the transport system (30,31). The translated command is thereby transferred from the requester program (13,15) to the host server (24).

The host server (24) oversees the execution of the command. The host server (24) may call data base intrinsics (22), operating system intrinsics (21) or remote procedure intrinsics (23) to execute the command.

When the host server (24) is started in response to a request originated by an application (11), host server (24) allocates a certain amount of memory on the host computer (20) for use by the application

EP 0 371 229 A3

(11) as a scratch pad. Through the use of remote procedures (23) the application (11) may store information to the scratch pad, modify data in the scratch pad and retrieve data from the scratch pad.

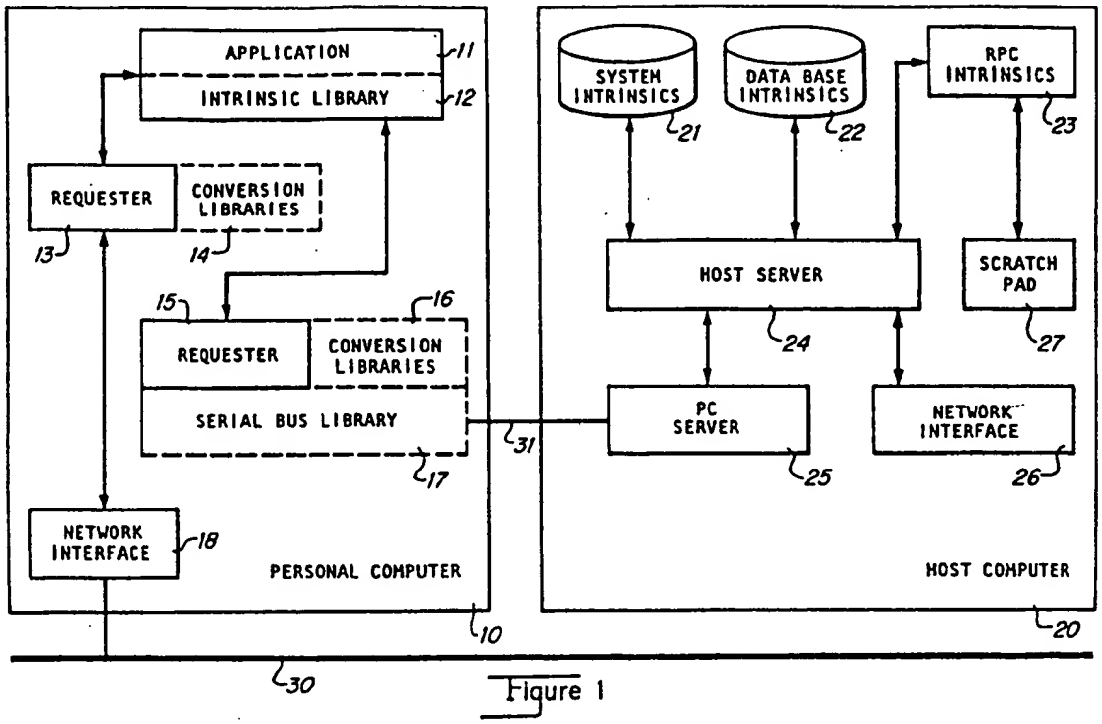

Figure 1

European Patent Office

EUROPEAN SEARCH REPORT

Application Number
EP 89117755

DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (Int. Cl.5)
X	EP-A-0193933 (WANG LABORATOIRIES INC.) * the whole document *	1-3,5-7	G06F15/16 G06F15/40 G06F9/46
X	EP-A-0006216 (IBM) * the whole document *	1,2,5,6	
X	JOURNAL OF TELECOMMUNICATION NETWORKS vol. 2, no. 3, 1983, pages 283-294, Rockville, Maryland, US; WANG et al.: "An application protocol for networkwide database access" * pages 285-287, paragraph 4.1; pages 291-293, paragraph 4.3 *	1,2,5,6	
X	COMMUNICATIONS OF THE ACM vol. 31, no. 3, March 1988, pages 258-273, New York, NY, US; NOTKIN et al.: "Interconnecting heterogeneous computer systems" * pages 260-264, paragraph "Remote procedure call"; pages 266-268, paragraph "Remote computation" *	1,2,5,6	
A	EP-A-0205948 (IBM) * the whole document *	1	

TECHNICAL FIELDS SEARCHED (Int. Cl.5)

G06F15/16
G06F15/40
G06F9/46
G06F9/40

The present search report has been drawn up for : claims 1-3,5-7,

Place of search	Date of completion of the search	Examiner
BERLIN	23.10.1989	DURAND, J.

CATEGORY OF CITED DOCUMENTS

X : particularly relevant if taken alone
Y : particularly relevant if combined with another document of the same category
A : technological background
O : non-written disclosure
P : intermediate document

T : theory or principle underlying the invention
E : earlier patent document, but published on, or after the filing date
D : document cited in the application
L : document cited for other reasons

& : member of the same patent family, corresponding document

EPO FORM 1503 03.82 (P0401)

European Patent Office

CLAIMS INCURRING FEES

The present European patent application comprised at the time of filing more than ten claims.

☐ All claims fees have been paid within the prescribed time limit. The present European search report has been drawn up for all claims.

☐ Only part of the claims fees have been paid within the prescribed time limit. The present European search report has been drawn up for the first ten claims and for those claims for which claims fees have been paid,

namely claims:

☐ No claims fees have been paid within the prescribed time limit. The present European search report has been drawn up for the first ten claims.

X LACK OF UNITY OF INVENTION

The Search Division considers that the present European patent application does not comply with the requirement of unity of invention and relates to several inventions or groups of inventions,
namely:

1. claims 1-3,5-7
   Executing a remote procedure or program on a host

2. claims 4,8
   Checking software version compatibility between requester and server

☐ All further search fees have been paid within the fixed time limit. The present European search report has been drawn up for all claims.

☐ Only part of the further search fees have been paid within the fixed time limit. The present European search report has been drawn up for those parts of the European patent application which relate to the inventions in respect of which search fees have been paid,

namely claims:

☒ None of the further search fees has been paid within the fixed time limit. The present European search report has been drawn up for those parts of the European patent application which relate to the invention first mentioned in the claims,

namely claims: 1-3,5-7